NAME

mps - LSI Fusion-MPT 2 IT/IR 6Gb/s Serial Attached SCSI/SATA driver

SYNOPSIS

To compile this driver into the kernel, place these lines in the kernel configuration file:

```
device pci
device scbus
device mps
```

The driver can be loaded as a module at boot time by placing this line in loader.conf(5):

```
mps_load="YES"
```

DESCRIPTION

The **mps** driver provides support for Broadcom Ltd./Avago Tech (LSI) Fusion-MPT 2 IT/IR SAS controllers and WarpDrive solid state storage cards.

HARDWARE

These controllers are supported by the **mps** driver:

- Broadcom Ltd./Avago Tech (LSI) SAS 2004 (4 Port SAS)
- Broadcom Ltd./Avago Tech (LSI) SAS 2008 (8 Port SAS)
- Broadcom Ltd./Avago Tech (LSI) SAS 2108 (8 Port SAS)
- Broadcom Ltd./Avago Tech (LSI) SAS 2116 (16 Port SAS)
- Broadcom Ltd./Avago Tech (LSI) SAS 2208 (8 Port SAS)
- Broadcom Ltd./Avago Tech (LSI) SAS 2308 (8 Port SAS)
- Broadcom Ltd./Avago Tech (LSI) SSS6200 Solid State Storage
- Intel Integrated RAID Module RMS25JB040
- Intel Integrated RAID Module RMS25JB080
- Intel Integrated RAID Module RMS25KB040
- Intel Integrated RAID Module RMS25KB080

CONFIGURATION

In all tunable descriptions below, X represents the adapter number.

To disable MSI interrupts for all **mps** driver instances, set this tunable value in loader.conf(5):

```
hw.mps.disable_msi=1
```

To disable MSI interrupts for a specific **mps** driver instance, set this tunable value in loader.conf(5):

dev.mps.X.disable_msi=1

To disable MSI-X interrupts for all **mps** driver instances, set this tunable value in loader.conf(5):

hw.mps.disable_msix=1

To disable MSI-X interrupts for a specific **mps** driver instance, set this tunable value in loader.conf(5):

dev.mps.X.disable_msix=1

To set the maximum number of DMA chains allocated for all adapters, set this tunable in loader.conf(5):

hw.mps.max_chains=NNNN

To set the maximum number of DMA chains allocated for a specific adapter, set this tunable in loader.conf(5):

dev.mps.X.max_chains=NNNN

The default max_chains value is 16384.

The current number of free chain frames is stored in the dev.mps.X.chain_free sysctl(8) variable.

The lowest number of free chain frames seen since boot is stored in the dev.mps.X.chain_free_lowwater sysctl(8) variable.

The number of times that chain frame allocations have failed since boot is stored in the dev.mps.X.chain_alloc_fail sysctl(8) variable. This can be used to determine whether the max_chains tunable should be increased to help performance.

The current number of active I/O commands is shown in the dev.mps.X.io_cmds_active sysctl(8) variable.

To set the maximum number of pages that will be used per I/O for all adapters, set this tunable in loader.conf(5):

hw.mps.max_io_pages=NNNN

To set the maximum number of pages that will be used per I/O for a specific adapter, set this tunable in loader.conf(5):

dev.mps.X.max_io_pages=NNNN

The default max_io_pages value is -1, meaning that the maximum I/O size that will be used per I/O will be calculated using the IOCFacts values stored in the controller. The lowest value that the driver will use for max_io_pages is 1, otherwise IOCFacts will be used to calculate the maximum I/O size. The smaller I/O size calculated from either max_io_pages or IOCFacts will be the maximum I/O size used by the driver.

The highest number of active I/O commands seen since boot is stored in the dev.mps.X.io_cmds_highwater sysctl(8) variable.

Devices can be excluded from **mps** control for all adapters by setting this tunable in loader.conf(5):

hw.mps.exclude_ids=Y

Y represents the target ID of the device. If more than one device is to be excluded, target IDs are separated by commas.

Devices can be excluded from **mps** control for a specific adapter by setting this tunable in loader.conf(5):

dev.mps.X.exclude_ids=Y

Y represents the target ID of the device. If more than one device is to be excluded, target IDs are separated by commas.

The adapter can issue the **StartStopUnit** SCSI command to SATA direct-access devices during shutdown. This allows the device to quiesce powering down. To control this feature for all adapters, set the

hw.mps.enable_ssu

tunable in loader.conf(5) to one of these values:

0 Do not send SSU to either HDDs or SSDs.

1 Send SSU to SSDs, but not to HDDs. This is the default value.

2 Send SSU to HDDs, but not to SSDs.

3 Send SSU to both HDDs and SSDs.

To control this feature for a specific adapter, set this tunable value in loader.conf(5):

dev.mps.X.enable_ssu

The same set of values are valid as when setting this tunable for all adapters.

SATA disks that take several seconds to spin up and fail the SATA Identify command might not be discovered by the driver. This problem can sometimes be overcome by increasing the value of the spinup wait time in loader.conf(5) with the

hw.mps.spinup_wait_time=NNNN

tunable. NNNN represents the number of seconds to wait for SATA devices to spin up when the device fails the initial SATA Identify command.

Spinup wait times can be set for specific adapters in loader.conf(5): with the

dev.mps.X.spinup_wait_time=NNNN

tunable. NNNN is the number of seconds to wait for SATA devices to spin up when they fail the initial SATA Identify command.

The driver can map devices discovered by the adapter so that target IDs corresponding to a specific device persist across resets and reboots. In some cases it is possible for devices to lose their mapped IDs due to unexpected behavior from certain hardware, such as some types of enclosures. To overcome this problem, a tunable is provided that will force the driver to map devices using the Phy number associated with the device. This feature is not recommended if the topology includes multiple enclosures/expanders. If multiple enclosures/expanders are present in the topology, Phy numbers are repeated, causing all devices at these Phy numbers except the first device to fail enumeration. To control this feature for all adapters, set the

hw.mps.use_phy_num

tunable in loader.conf(5) to one of these values:

-1 Only use Phy numbers to map devices and bypass the driver’s mapping logic.

0 Never use Phy numbers to map devices.

1 Use Phy numbers to map devices, but only if the driver's mapping logic fails to map the device that is being enumerated. This is the default value.

To control this feature for a specific adapter, set this tunable value in loader.conf(5):

dev.mps.X.use_phy_num

The same set of values are valid as when setting this tunable for all adapters.

DEBUGGING

Driver diagnostic printing is controlled in loader.conf(5) by using the global *hw.mps.debug_level* and per-device *dev.mps.X.debug_level* tunables. One can alter the debug level for any adapter at run-time using the sysctl(8) variable *dev.mps.X.debug_level*.

All *debug_level* variables can be named by either an integer value or a text string. Multiple values can be specified together by either ORing the integer values or by providing a comma-separated list of names. A text string prefixed by "+" adds the specified debug levels to the existing set, while the prefix "-" removes them from the existing set. The current *debug_level* status is reported in both formats for convenience. The following levels are available:

Flag	*Name*	*Description*
0x0001	info	Basic information (enabled by default)
0x0002	fault	Driver faults (enabled by default)
0x0004	event	Controller events
0x0008	log	Logging data from controller
0x0010	recovery	Tracing of recovery operations
0x0020	error	Parameter errors and programming bugs
0x0040	init	System initialization operations
0x0080	xinfo	More detailed information
0x0100	user	Tracing of user-generated commands (IOCTL)
0x0200	mapping	Tracing of device mapping
0x0400	trace	Tracing through driver functions

SEE ALSO

cam(4), cd(4), ch(4), da(4), mpr(4), mpt(4), pci(4), sa(4), scsi(4), targ(4), loader.conf(5), mpsutil(8), sysctl(8)

HISTORY

The **mps** driver first appeared in FreeBSD 9.0.

AUTHORS

The **mps** driver was originally written by Scott Long <*scottl@FreeBSD.org*>. It has been improved and tested by LSI Corporation, Avago Technologies (formerly LSI), and Broadcom Ltd. (formerly Avago).

This manual page was written by Ken Merry <*ken@FreeBSD.org*> with additional input from Stephen McConnell <*slm@FreeBSD.org*>.